Ordonnancement de transmissions périodiques

N. Dridi, M.J. Teixeira

IRIA
CENTRE DE ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P. 105
78153 Le Chesnay Cedex
France
Tel. (3) 954 90 20

Rapports de Recherche

N° 407

ORDONNANCEMENT DE TRANSMISSIONS PÉRIODIQUES

Najoua DRIDI
Mario TEIXEIRA

Mai 1985

ORDONNANCEMENT DE

TRANSMISSIONS PERIODIQUES

DRIDI N.*

TEIXEIRA M.**

* INRIA : Projet SAGEP, Château du Montet, 54506 VANDOEUVRE
Tél. : (8) 355-15-45

** INRIA : Projet SCORE, Domaine de Voluceau, ROCQUENCOURT - BP 105,
78153 LE CHESNAY CEDEX Tél. : (3) 954-90-20

ABSTRACT

In this paper we consider n periodic tasks which have to be executed on m sites. Each task is composed of one sender action and one or more receiver actions. Each action is executed on a given site an its execution time is known. Sender and receiver actions belonging to the same task cannot be executed simultaneously ; all sender actions are executed on the same site.

Finally, we suppose that actions are not preempted during execution. We describe a polynomial algorithm that minimizes the execution period of a given set of tasks.

RESUME

Dans ce papier, nous considérons le cas de n tâches périodiques qui doivent être exécutées sur m sites. Chaque tâche est composée d'une action d'émission et d'une ou plusieurs actions de réception. Pour chaque action la durée d'exécution est connue ainsi que le site sur lequel a lieu l'exécution.

Nous supposons que toutes les émissions ont lieu à partir du même site. Nous n'admettons pas de chevauchement dans le temps entre émission et réception d'actions concernant la même tâche.

Enfin, nous supposons que l'exécution d'une action est non-préemptible. Nous proposons un algorithme polynomial qui minimise la période d'exécution des tâches.

1. CONTEXTE DU TRAVAIL

On appelle tâche un ensemble d'émissions et de réceptions relatives au même message.
On considère un ensemble de tâches périodiques qui s'exécutent sur un système temps réel réparti. Le système est composé d'un nombre m de sites dotés d'une puissance de calcul. Ces sites sont mono-processeurs et inter-liés par un réseau local à haut débit.

Dans ce qui suit, nous allons considérer le cas particulier des tâches de communication périodiques entre sites éloignés. Comme exemple de ce type d'application, nous pourrons citer les liaisons téléphoniques, la transmission des valeurs des paramètres surveillés par des capteurs, etc.

Une tâche est composée d'actions qui sont soit des émissions soit des réceptions de messages, et leurs traitements. Chaque action a un site spécifié par l'utilisateur pour son exécution. L'attachement des actions aux sites peut être déterminé en fonction des contraintes physiques de l'installation ou des choix stratégiques. Par exemple pour le premier cas, la proximité entre un site et un capteur de température situé dans un four est un facteur décisif pour le rattachement du capteur à ce site ; de même, dans le second cas, un centre de contrôle installé à un endroit précis peut avoir besoin de communications téléphoniques permanentes avec d'autres sites éloignés.

Les durées d'exécution des émissions et des réceptions sont connues. La durée d'exécution d'une émission comprend l'accès au médium et le placement du message dans un tampon du site récepteur ; le message reste disponible pour être traité par l'action de réception de la tâche. On suppose la taille des tampons assez large pour éviter des problèmes de congestion. Les durées des actions sont quelconques et leurs exécutions non-préemptibles, i.e. une fois lancée, une action n'est pas interrompue.

Deux modèles typiques sont pris en compte :

- 1 émetteur / 1 récepteur (Fig. 1)
- 1 émetteur / plusieurs récepteurs (Fig. 2)

Ces modèles peuvent être représentés par des graphes orientés dans lesquels à chaque sommet, on associe une action :

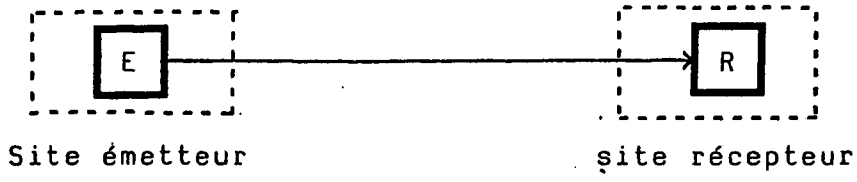


Fig. 1

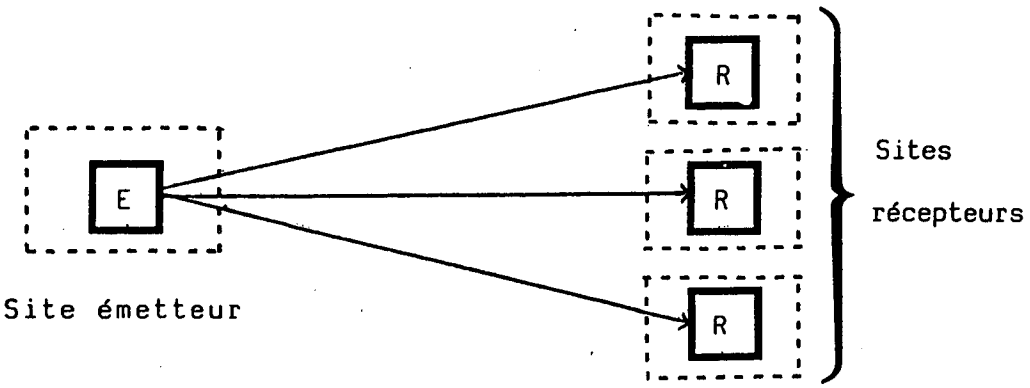


Fig 2.

Nous considérons que toutes les tâches doivent s'exécuter avec la même périodicité. L'énoncé du problème est le suivant : est-il possible de trouver un algorithme polynomial qui minimise la période d'exécution d'un ensemble de tâches spécifiées par l'utilisateur ?

2. QUELQUES REMARQUES ET RAPPELS DES RESULTATS

Du fait que l'exécution des tâches est périodique, les grâphes qui représentent les modèles peuvent être pensés comme étant cycliques. Par conséquent, l'ordre d'exécution des émissions et réceptions est sans importance dans un intervalle de longueur égale à la période P. Ce qui nous intéresse c'est de ré-

péter des configurations dans le temps sur chacun des sites. Dans le cas des actions non-préemptibles, cela veut dire que les débuts des exécutions d'une action pour deux réalisations successives, soient toujours écartés de P unités de temps. Cette remarque implique qu'un observateur qui désire voir l'éxécution d'une réalisation complète de chacune des tâches, doit observer le système, dans le pire des cas, pendant un intervalle de longueur 2P, même si toutes les tâches s'exécutent avec une période P.

Nous nous imposons comme contraintes supplémentaires, le fait qu'on ne puisse pas commencer une nouvelle exécution de tâche si on n'a pas encore fini l'exécution de la réalisation antérieure. Cela se traduit dans les modèles étudiés par l'inexistence de "chevauchements" entre actions d'émissions et de réceptions d'une même tâche.

Finalement, nous pouvons remarquer que les actions peuvent "boucler" leurs exécutions, comme indiqué par la figure 3. Cela ne les empêche pas d'avoir des exécutions non-préemptibles et constitue une hypothèse compatible avec les autres remarques.

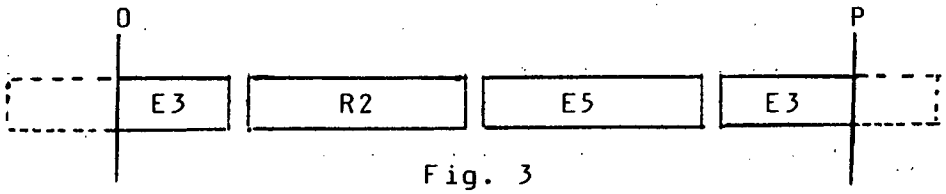


Fig. 3

Notre formulation du problème présente beaucoup de similarités avec le problème de minimisation du temps de fin d'un ensemble de tâches dans un "open-shop"* dont les actions sont non-préemptibles.

Cependant, pour voir la différence entre les deux, prenons le contre-exemple suivant :

- 4 tâches et 5 sites.
- durée des émissions = 2, durée des réceptions = 5
- attribution des actions aux sites :

T1 : E1 ⟶ S1 R1 ⟶ S2
T2 : E2 ⟶ S1 R2 ⟶ S3
T3 : E3 ⟶ S1 R3 ⟶ S4
T4 : E4 ⟶ S1 R4 ⟶ S5

* On appelle "open-shop" un job-shop dans lequel l'ordre des opérations est libre.

Le temps minimal de fin pour l'ensemble est T = 9, obtenu, par exemple par la figure 4.

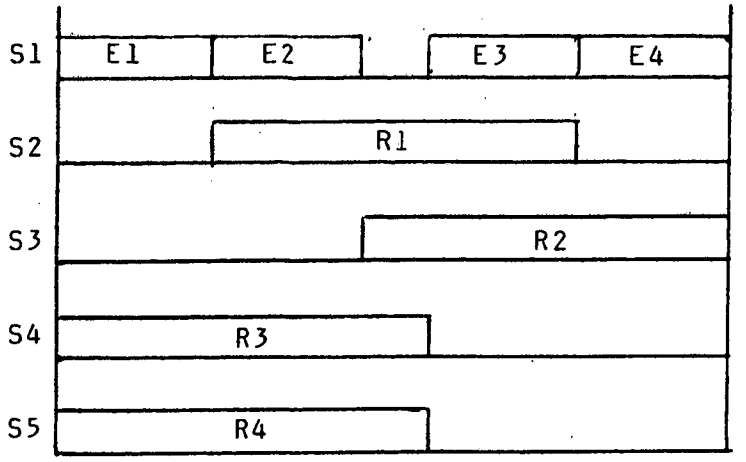
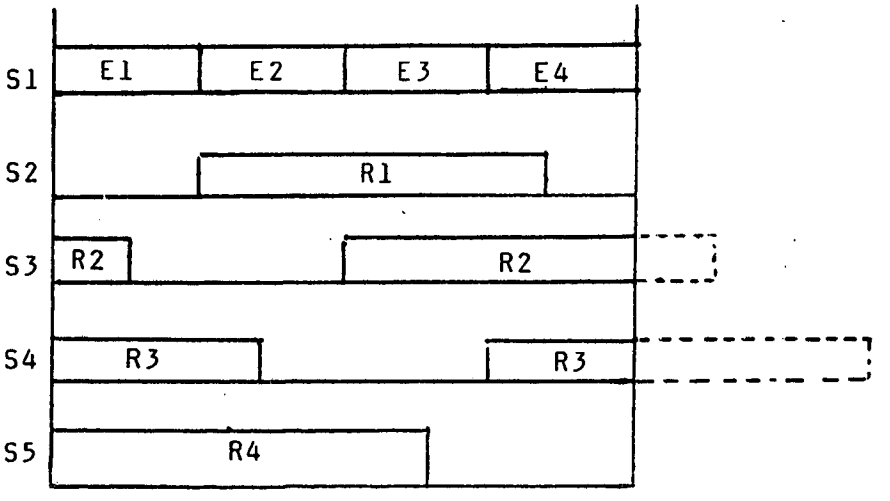


Fig. 4

La période minimale pour l'ensemble est P=8.



Les problèmes qui consistent à minimiser le temps de fin d'exécution d'un ensemble de tâches dans un "open-shop" sont NP-complets pour des systemes de plus de 2 sites pour des tâches ne contenant qu'un émetteur et qu'un récepteur (cho 81). La complexité de notre problème reste ouverte. Pour certains cas particuliers, des solutions optimales peuvent être trouvées, ou des bornes supérieures ont été établies (Teixeira 85).

Ici nous présentons une solution optimale polynomiale pour le cas où tous les émetteurs (respectivement tous les récepteurs) sont situés sur un site et les récepteurs (respectivement les émetteurs) sur les autres sites du système.
L'algorithme s'applique aussi au cas où il n'y a que deux sites dans le système (1 émetteur / 1 récepteur ou encore 2 émetteurs-récepteurs) et simplifie l'algorithme de minimisation du temps de fin d'un "open-shop" décrit dans (Gonzalez 76) en obtenant la même valeur optimale.

3. DONNEES DU PROBLEME

Nous considérons n tâches périodiques à éxécuter sur m sites. Nous supposons qu'une tâche ne peut avoir qu'un seule émission, mais peut avoir plusieurs réceptions sur des sites différents du site d'émission.

Nous nous intéressons au cas particulier où toutes les émissions (ou toutes les réceptions) ont lieu sur un même site. Les durées des émissions et des réceptions sont connues.

(1) On n'admet pas de chevauchement dans le temps entre l'intervalle d'émission et celui de réception d'une même tâche.

Nous acceptons le fait de "boucler" une exécution, en d'autres termes, une tâche peut chevaucher deux périodes (voir Fig. 3).
Le but recherché est de trouver un ordonnancement des n tâches sur les m sites qui minimise la période.

REMARQUE : La période minimale :

1) sera supérieure ou égale au temps nécessaire pour émettre ou recevoir les messages sur chaque site.
2) sera supérieure ou égale au maximum de la somme d'une durée d'émission et d'une durée de réception pour chaque tâche.

La borne minimale de la période s'écrit :

(2)
$$P_{min} = \max\left[\max_{j=1,m} (\text{total de la durée des tâches sur } S_j) \; ; \; \max_{i=1,n} (\text{temps minima séparant le début d'émission de la fin de réception de la tâche } i) \right]$$

DEFINITION

On appelle ordonnancement admissible, un ordonnancement où toutes les tâches vérifient l'hypothèse (1).

NOTATIONS

On note :

$t_i^0(j)$: l'instant de début d'exécution d'une action de la tâche i sur le site S_j.

$t_i^1(j)$: l'instant de fin d'exécution d'une action de la tâche i sur le site S_j.

4. ORDONNANCEMENT OPTIMAL

THEOREME

Il existe un ordonnancement admissible, qui conduit à une période égale à P_{min}.

DEMONSTRATION

Soit S_{j_0} le site où sont situées toutes les émissions, soit Sj un site où sont situées toutes les émissions, soit S_j un site différent de S_{j_0}. Plaçons les émissions et les réceptions à partir de l'instant zéro sur chacun des sites du système en commencant l'exécution d'une action dès que l'action précédente est finie et en gardant le même ordre (quelconque) sur tous les sites.

Soient :

$$a_k(j) = t_k^1(j_0) - t_k^0(j)$$

$$b_k(j) = t_k^1(j) - t_k^0(j_0)$$

Remarquons qu'une condition nécessaire et suffisante de chevauchement entre l'intervalle d'émission d'une tâche et celui de sa réception sur un site S_j est :

$$a_k(j) > 0 \quad \text{et} \quad b_k(j) > 0$$

Supposons que nous sommes dans ce cas.

Soit k_0 la tâche telle que :

(3) $$a_{k_0}(j) = \max_k \left(a_k(j)\right)$$

Nous translatons tous les instants d'exécution sur le site S_j d'une valeur égale à $a_{k_0}(j)$. Cela veut dire qu'après translation, la tâche k_0 commencera sa réception juste après avoir fini son émission :

$$t^1_{k_0}(j_0) - t^{0'}_{k_0}(j) = t^1_{k_0}(j_0) - \left(t^0_{k_0}(j) + a_{k_0}(j)\right) = 0$$

avec $t^{0'}_{k_0}(j)$: l'instant de début d'exécution de l'action de la tâche k_0 sur S_j après translation.

Pour toute autre réception concernant une tâche k, placée avant k_0 sur S_j et qui donc ne "bouclera" pas son exécution, nous avons :

$$t^1_k(j_0) - t^{0'}_k(j) = t^1_k(j) - t^0_k(j) - a_{k_0}(j)$$

avec (3) nous aurons :

$$t^1_k(j_0) - t^0_k(j) \leq t^1_k(j_0) - t^0_k(j) - a_k(j) = 0$$

i.e. l'émission de la tâche k finie avant le début de sa réception sur S_j.

Examinons maintenant les actions des tâches k placées après k_0.

Les exécutions sur S_j qui "bouclent" complétement vérifient (1) puisqu'elle vont s'exécuter avant k_0 et

$$t^{0'}_{k_0}(j) = t^1_{k_0}(j_0) \leq t^0_k(j_0)$$

ainsi leurs exécutions sur S_j finissent avant leurs débuts d'émissions sur S_{j_0}.

Les exécutions sur S_j placées après k_0 et qui, après translation, restent placées après k_0 vérifient aussi (1) :

$$t^1_k(j_0) - t^{0'}_k(j) = t^1_k(j_0) - t^0_k(j) - a_{k_0}(j)$$

$$\leq t^1_k(j_0) - t^0_k(j) - a_k(j) = 0$$

d'où

$$t^1_k(j_0) \leq t^{0'}_k(j)$$

L'algorithme consiste à, s'il y a chevauchement entre l'intervalle d'émission et celui de réception sur un site Sj d'une tâche quelconque, translater tous les instants d'éxécution de Sj d'une valeur égale à $\max_k \left(a_k(j)\right)$ ce qui revient aussi à translater tous les instants d'exécution de Sj de $-\max_k \left(b_k(j)\right)$ à cause de la périodicité des tâches. Après translation, on "boucle" les exécutions qui se situent en dehors de $\left[0,P_{min}\right]$. La complexité d'un algorithme est O(nm).

CONCLUSION

Nous venons de donner un algorithme polynomial pour le problème comportant un site émetteur et plusieurs récepteurs, les temps d'émission et réception sont quelconques.

Deux exemples particuliers ont été également traités et sont en cours de rédaction, il s'agit du cas de 3 sites qui sont indifféremment sites d'émission et de réception, et du cas de groupe de plusieurs sites, le ler groupe étant composé uniquement de sites d'émission et le second, de sites de réception (Dridi 85). Dans ce dernier cas, on fait cependant l'hypothèse que toutes les durées d'émission sont égales ainsi que toutes les durées de réception. Le cas général où tous les sites sont indifféremment sites d'émission et de réception et où les temps sont quelconques reste ouvert..

BIBLIOGRAPHIE

1. **GONZALEZ 1976** : Open-shop Scheduling to Minimize Finish time. T.GONZALEZ and S. SAHNI. JACM, vol. 23, nº 4, Octobre 1976, pp 665-679.

2. **TEIXEIRA 1985** : Thèse de Docteur Ingénieur (en préparation).

3. CHO 1981 : Preemptive Scheduling of independent jobs with release and due times and open, flow and job shops.
Y. CHO and S. SAHNI. Op. research, Vol 29, May-June 1981, Pg. 511-522.

4. DRIDI 1985 : Thèse de 3ème cycle (en préparation).

PROGRAMME

```
c Programme d'ordonnancement de n taches periodiques sur m sites,minimisant
c la periode,en supposant qu'il y a soit un seul emetteur soit un
c seul recepteur.
c
c*************************************************************************
c************************** ENTREES **************************************
c*************************************************************************
c
c n         : Nombre de taches
c m         : Nombre de sites
c j0        : Numero du site ,qui est seul emetteur ou seul recepteur parmi
c             tous les sites.
c d(i,j),
c i=1,n
c j=1,m     : Duree d'execution de la tache i sur le site j.
c
c*************************************************************************
c************************** SORTIES **************************************
c*************************************************************************
c
c pmin      : Valeur de la periode minimale.
c t0,t1(i,j),
c i=1,n,
c j=1,m     : Temps de debut (resp.fin) d'execution de la tache placee au rang
c             i sur le site j.
c ip(i,j),
c i=1,n
c j=1,m     : Numero du produit place au rang i sur le site j.
c
c*************************************************************************
c*************************************************************************
c
      subroutine pemini(j0,n,m,d,pmin,t0,t1,ip)
      dimension d(50,50),t0(50,50),t1(50,50),ip(50,50),i1(50),i2(50)
      dimension l(50)
      eps=1.e-3
      do 1 i=1,n
      dd=d(i,1)
      d(i,1)=d(i,j0)
      d(i,j0)=dd
 1    continue
      do 5 i=1,n
      i1(i)=0
      i2(i)=0
      do 10 j=1,m
      ip(i,j)=0.
      t0(i,j)=0.
      t1(i,j)=0.
 10   continue
 5    continue
```

```
c-----------------------------------------------------------------------
c  CALCUL DE LA PERIODE MINIMALE pmin.
c-----------------------------------------------------------------------
      pmin=0.
      do 6 j=1,m
      dd=0.
      do 7 i=1,n
      dd=dd+d(i,j)
 7    continue
      pmin=amax1(pmin,dd)
 6    continue
      do 8 i=1,n
      dd=0.
      do 9 j=2,m
      dd=amax1(dd,d(i,1)+d(i,j))
 9    continue
      pmin=amax1(pmin,dd)

 8    continue
c-----------------------------------------------------------------------
c  PLACEMENT DES TACHES SUR LES DIFFERENTS SITES
c-----------------------------------------------------------------------
      do 2 j=1,m
      x=0.
      do 15 i=1,n
      t0(i,j)=x
      t1(i,j)=x+d(i,j)
      ip(i,j)=i
      x=t1(i,j)
 15   continue
 2    continue
c-----------------------------------------------------------------------
c  MODIFICATION DU PLACEMENT DES TACHES QUAND IL Y A CHEVAUCHEMENT ENTRE
c  EMISSION ET RECEPTION D'UNE MEME TACHE.
c-----------------------------------------------------------------------
      do 100 j=2,m
      ee=0.
      do 40 i=1,n
      if((t1(i,j).le.t0(i,1)+eps).or.(t0(i,j).ge.t1(i,1)-eps))go to 40
      e=t1(i,1)-t0(i,j)
      ee=amax1(ee,e)
 40   continue
      if(ee.le.eps)go to 100
      do 45 k=1,n
      t0(k,j)=t0(k,j)+ee
      t1(k,j)=t1(k,j)+ee
 45   continue
      l(j)=l(j)+e
      do 110 i=1,n
      if(t1(i,j).le.pmin+eps)go to 110
      e=t1(i,j)-pmin
      t1(i,j)=e
      if(i.ge.n)go to 110
      do 115 ii=i+1,n
      t0(ii,j)=t1(ii-1,j)
      t1(ii,j)=t0(ii,j)+d(ii,j)

115   continue
110   continue
100   continue
      do 60 i=1,n
      tt=t0(i,1)
      t0(i,1)=t0(i,j0)
      t0(i,j0)=tt
      tt=t1(i,1)
      t1(i,1)=t1(i,j0)
      t1(i,j0)=tt
      ii=ip(i,1)
      ip(i,1)=ip(i,j0)
      ip(i,j0)=ii
      dd=d(i,1)
      d(i,1)=d(i,j0)
      d(i,j0)=dd
 60   continue
      return
      end
```

EXEMPLE

****
* DONNEES *
****

NOMBRE DE TACHES: 6,NOMBRE DE SITES: 4

SITE 1 EMETTEUR

TACHE	DUREE D'EXECUTION
1	3.0000
2	4.0000
3	6.0000
4	2.0000
5	5.0000
6	5.0000

SITE 2 RECEPTEUR

TACHE	DUREE D'EXECUTION
1	5.0000
2	3.0000
3	6.0000
4	5.0000
5	5.0000
6	4.0000

SITE 3 RECEPTEUR

TACHE	DUREE D'EXECUTION
1	4.0000
2	3.0000
3	5.0000
4	11.0000
5	5.0000
6	4.0000

SITE 4 RECEPTEUR

TACHE	DUREE D'EXECUTION
1	3.0000
2	11.0000
3	17.0000
6	3.0000

****
* RESULTATS *
****

VALEUR DE LA PERIODE MINIMALE : 34.0000

SITE 1

TACHE	Tps DEBUT D'EXECUTION	Tps FIN D'EXECUTION
1	0.0000	3.0000
2	3.0000	7.0000
3	7.0000	13.0000
4	13.0000	15.0000
5	15.0000	20.0000
6	20.0000	25.0000

SITE 2

TACHE	Tps DEBUT D'EXECUTION	Tps FIN D'EXECUTION
1	5.0000	10.0000
2	10.0000	13.0000
3	13.0000	19.0000
4	19.0000	24.0000
5	24.0000	29.0000
6	29.0000	33.0000

SITE 3

TACHE	Tps DEBUT D'EXECUTION	Tps FIN D'EXECUTION
1	6.0000	10.0000
2	10.0000	13.0000
3	13.0000	18.0000
4	18.0000	29.0000
5	29.0000	34.0000
6	34.0000	4.0000

SITE 4

TACHE	Tps DEBUT D'EXECUTION	Tps FIN D'EXECUTION
6	1.0000	4.0000
1	4.0000	7.0000
2	7.0000	18.0000
3	18.0000	1.0000